IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 2, 2021, accepted February 16, 2021, date of publication February 24, 2021, date of current version March 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3061716

3D Hand Pose Estimation via Graph-Based Reasoning

JAE-HUN SONG AND SUK-JU KANG, (Member, IEEE)
Department of Electronic Engineering, Sogang University, Seoul 04107, South Korea

Corresponding author: Suk-Ju Kang (sjkang@sogang.ac.kr)

This research was supported by Ministry of Culture, Sports and Tourism and Korea Creative Content Agency (Project Number: R2020040058), and the MSIT(Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2018-0-01421) supervised by the IITP (Institute of Information & communications Technology Planning & Evaluation).

ABSTRACT Hand pose estimation from a single depth image has recently received significant attention owing to its importance in many applications requiring human–computer interaction. The rapid progress of convolutional neural networks (CNNs) and technological advances in low-cost depth cameras have greatly improved the performance of the hand pose estimation method. Nevertheless, regressing joint coordinates is still a challenging task due to joint flexibility and self-occlusion. Previous hand pose estimation methods have limitations in relying on a deep and complex network structure without fully utilizing hand joint connections. A hand is an articulated object and consists of six parts that represent the palm and five fingers. The kinematic constraints can be obtained by modeling the dependency between adjacent joints. This paper proposes a novel CNN-based approach incorporating hand joint connections to features through both a global relation inference for the entire hand and local relation inference for each finger. Modeling the relations between the hand joints can alleviate critical problems for occlusion and self-similarity. We also present a hierarchical structure with six branches that independently estimate the position of the palm and five fingers by adding hand connections of each joint using graph reasoning based on graph convolutional networks. Experimental results on public hand pose datasets show that the proposed method outperforms previous state-of-the-art methods. Specifically, our method achieves the best accuracy compared to state-of-the-art methods on public datasets. In addition, the proposed method can be utilized for real-time applications with an execution speed of 103 fps in a single GPU environment.

INDEX TERMS 3D hand pose estimation, depth image, graph convolutional network.

I. INTRODUCTION

Hand pose estimation is the task of predicting the position and orientation of the palm and fingers when given volumetric data captured by a depth camera. It is an important research topic in virtual (or augmented) reality systems and gesture-based human-computer interaction systems [1], [2]. Although there have been many studies on improving the performance of hand pose estimation, it still remains a challenging task owing to the constraints from the physiology of the hands, such as the high degree of flexibility, occlusions, local self-similarity, and small hand area of the image and noise from the depth camera. In recent years, the emergence of commodity depth cameras with high-performance visual sensors such as Intel RealSense [3] and Microsoft Kinect [4] has made pose estimation much easier by solving the depth ambiguity issue, and most recent methods are largely based on depth images [5], [6].

Computer vision-based hand pose estimation methods can be categorized into three types [7]. The first is a generative (model-based) method [8]–[11] to predict the position of the hand joints using a 3D hand model constructed based on prior knowledge of the hand structure. The hand model exploits hand shape constraints but is vulnerable to accumulated estimation errors and difficult to apply in real-time owing to an excessive computational burden during the optimization process. The second is a discriminative (data-driven) method [12]–[17], which finds the position of the joints by learning directly from the dataset image, and is the most commonly used method for a hand pose estimation.

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

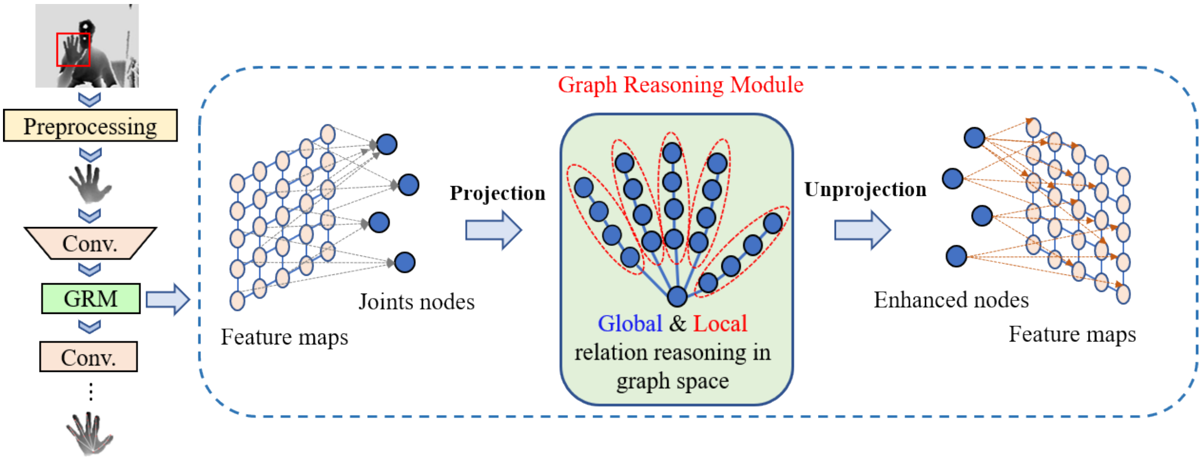

FIGURE 1. Main concept of the proposed method. After extracting the initial features from the input image through several convolutions, it is updated to include spatial connection information in the global feature of the hand and the local feature of fingers through relation reasoning using GCN-based GRM. The feature maps of the coordinate space are projected to the joint nodes of the graph space, and then reasoned node features are projected back to the coordinate space.

Typical methods include a random forest and convolutional neural networks (CNNs). The last hybrid method [18]–[20] combines the advantages of the generative method and the discriminative method, but requires a high computational cost increased by the generative method. This paper focuses on discriminative methods that improve the accuracy and efficiency of real-time applications.

Most state-of-the-art methods have recently been based on the CNN model owing to the high performance of deep learning technology. In addition, the performance has been greatly improved with the emergence of large public hand pose estimation datasets [21]–[23]. CNN model architectures can be divided into 2D and 3D approaches according to the type of input. The first 2D input method uses 2D CNNs and can be classified into global regression [12], [13], [24], [25], and local detection regression [14], [15], [17], [22], [26]–[28]. A local detection-based method that can use local spatial context information is applied more frequently than global regression methods. The second 3D input method uses a point cloud or voxel as an input to improve the performance. In general, 3D CNN-based methods [14], [16], [29]–[31] voxelize a depth image into a volume representation. These 3D CNN methods have a large number of parameters, thereby reducing the computational cost. For this reason, efficient 2D CNN-based methods are still being researched.

This paper proposes a novel network that applies a graph convolutional network (GCN) based graph reasoning module (GRM) to obtain features that contain more useful context information when applying a 2D CNN method. As shown in Fig. 1, the proposed method applies GRM to both the global feature network and the local feature network to process the relation reasoning for capturing spatial connection information between joint nodes in the graph space. Then, these reasoned node features are reflected to improve the features within the coordinate space. Regarding this procedure, GRM changes the extracted features to the features corresponding to each joint node, and then learns the connection information between joints through GCN-based graph reasoning. Compared with a CNN, a GCN is highly suitable for relation reasoning between the hand joints corresponding to the nodes because it can directly infer the relation between joint nodes. The features, including context information between hand joints, are mapped back to the features in the coordinate space to obtain the final enhanced features. We apply the GRM to the output feature of the encoder–decoder structure for global feature extraction, and construct a hierarchical architecture divided into palms and fingers (thumb, index, middle, ring, and little fingers) to enhance the local features, and apply the GRM to each finger branch. These enhanced global and local features improve the accuracy of the performance with the ground truth when estimating the joint coordinates in the final regression module.

Our contributions are summarized as follows:

1) We propose a new approach for graph reasoning by projecting features that are globally aggregated over the coordinated space into a graph space where relation reasoning can be efficiently computed. Graph reasoning is used to learn the inter-joint relation between joint nodes and includes the information in features to improve the accuracy when estimating the final coordinates. After reasoning, the reasoned features are distributed back to the coordinate space for the next tasks.

2) We propose an architecture that applies a GRM to enhance the extracted features for a hand pose estimation. To improve not only the global features corresponding to the entire hand information but also the local features corresponding to each finger, we designed a hierarchical model of six branches divided into the palm and five fingers.

3) We conducted extensive experiments on hand pose estimation datasets. Experimental results on three public hand pose estimation datasets show that the proposed method achieves a better performance than previous state-of-the-art methods in terms of accuracy and efficiency.

The remainder of this paper is organized as follows. Section II reviews studies related to 3D hand pose estimation.

Section III introduces the details of the proposed method. Section IV presents experimental results, including a self-comparison and a performance comparison with state-of-the-art methods, and the final section presents some concluding remarks regarding this research.

II. RELATED WORK

A. HIERARCHICAL MULTI-BRANCH ARCHITECTURE

The hand pose estimation problem can be subdivided into multiple tasks to handle the mapping from the input depth image to the hand joint coordinate output. The hierarchical network structure with multiple branches divides the hand joints into small subsets and extracts local pose features for each subset. Then, all local pose features are combined to better estimate the final global pose.

Madadi *et al.* [15] proposed a hierarchically structured CNN using six branches consisting of five-branch modeling of each finger and one-branch modeling of the palm orientation. After each local pose feature is extracted, these local features are concatenated to predict the coordinates of all joints. Besides, physical constraints are incorporated to the loss function to avoid unrealistic pose configurations.

Chen *et al.* [17] extracted regions from the feature maps of a CNN and generated more optimal and representative features for a hand pose estimation. These feature regions are then integrated hierarchically according to the topology of the hand joints through tree-like connections to regress the refined hand pose. This method can learn better features for hand pose estimation by incorporating guided information of previously estimated hand pose.

Zhou *et al.* [27] presented a three-branch network according to the functional importance of the finger usage. These three branches represent the thumb, index, and three other fingers, respectively. The core idea is to take advantage of the prior knowledge of the motion. Since the thumb and index finger play a more important role in the grasping, manipulation, while other fingers play an auxiliary role in most cases.

Du *et al.* [28] proposed two-branch networks of the palm and fingers. This method inspired from the multi-task mechanism. A flexible finger pose and relatively stable palm pose are expressed individually, and the accuracy is increased by sharing information with each other.

However, these methods have the weakness of directly estimating all finger poses without considering the finger kinematics. Because the fingers are connected by joints, a spatial dependency occurs between adjacent joints. Therefore, features including joint connection information can improve the accuracy of the joint coordinate estimation. We apply a GCN-based graph reasoning module to extract features that incorporate this connection information.

B. GRAPH-BASED REASONING

Graph-based methods [36]–[38] have been widely used in computer vision tasks and proven to be extremely effective for relation reasoning. In [36] and [37], the authors applied a CNN to the spectral domain based on a graph Laplacian. Kipf and Welling [38] first proposed the use of a GCN for semi-supervised classification. Since then, a GCN has been widely used in various tasks. Wang and Gupta [39] used a GCN to capture relations between objects in video recognition tasks, and Chen *et al.* [40] proposed the use of graph-based global reasoning networks. They also designed a global reasoning unit for reasoning between separate and distant regions. Liang *et al.* [43] used a GCN to enhance the local features on semantic segmentation and image classification tasks. In addition, the skeleton-based action recognition tasks [44], [45] utilized a GCN to significantly improve the accuracy and efficiency.

A CNN-based method using only a pixel grid image ignores the relation between disjointed joints. Therefore, we apply GCN-based relation reasoning modules inspired by previous graph-based tasks [40]–[43], which model the relation between regions. These modules allow features that include better structural connection information when conducting a 3D hand pose estimation.

III. PROPOSED METHOD

A. OVERALL NETWORK ARCHITECTURE

In preprocess stage, we assume that the hand is the nearest object to the depth camera. We can detect the hand using the depth information. We cropped the hand region and resized it to 96 × 96. Then this resized depth map is entered into our architecture.

Fig. 2 shows the overall architecture of our proposed 3D hand pose estimation method applying graph-based global and local relation reasoning modules. The proposed architecture can be divided into three major parts according to their function. The first part is the initial feature extraction module and refinement, where we get all initial joint features from an input depth image and improves global features by applying global GRM. The second part is the local feature refinement, where we divide the extracted features into six branches corresponding to the palm and five fingers, and improves the local features of the finger branches. The reasons why we chose hierarchical structure are that it can be simplified by dividing into sub-tasks and the five-finger features can be enhanced again by applying local GRM. The third part is regression, where we combine six feature maps from the palm and fingers, and eventually predicts the 3D coordinates of the entire hand joint.

B. GLOBAL FEATURE EXTRACTION AND REFINEMENT

1) INITIAL FEATURE EXTRACTION

This network takes the depth image as an input with a spatial size of 96 × 96 and outputs the feature maps with a spatial size of 12 × 12. To extract features from an input image, we modify and use the highly efficient ResNet-50 [46] as a backbone network. In addition, to obtain more feature information, we constructed an upsampling decoder that combines

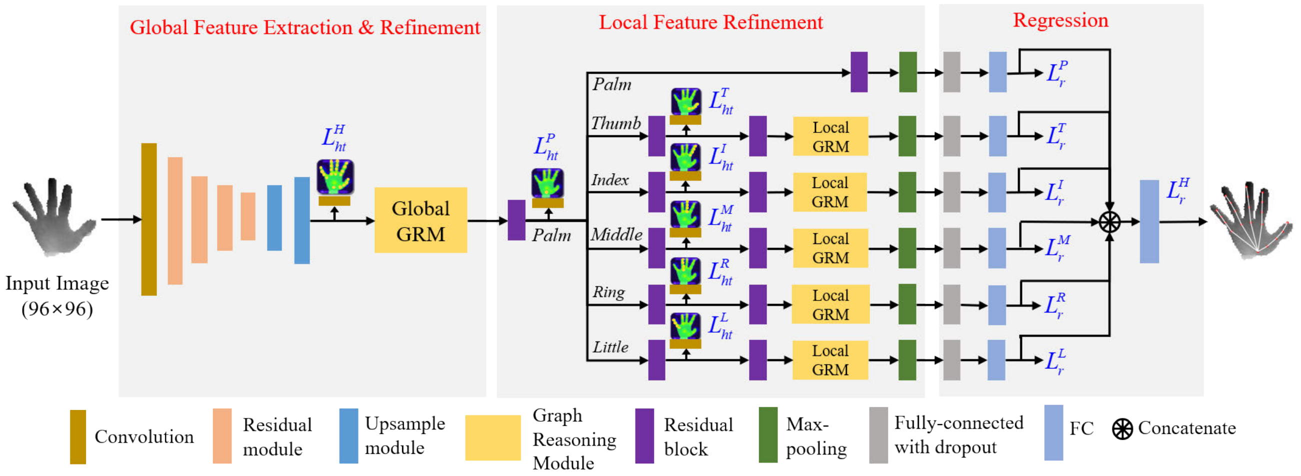

FIGURE 2. Overall architecture of the proposed method with graph-based global and local relation reasoning modules.

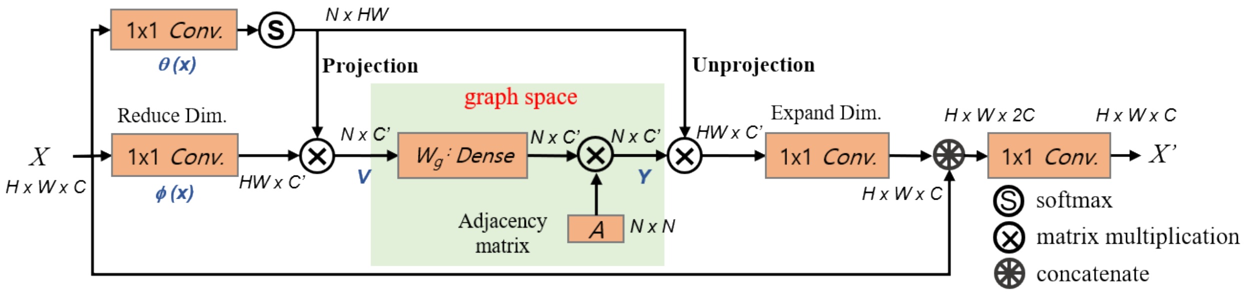

FIGURE 3. Architecture of one graph reasoning module (GRM) by taking the convolution feature map of H x W x C as an input. It consists of four convolutions and one linear layer, two convolutions for a dimension reduction and expansion, and one convolution for generating the projections between the coordinate and graph spaces. The linear layer encodes the features as graph nodes, and node features are then evolved through matrix multiplication with adjacency a matrix A.

previous features using simple bilinear interpolation and CNNs. Inspired by CrossInfoNet [28], we apply heat map guidance, which is as a constraint that guides the feature extraction to obtain better features. These initial features are fed into the global GRM.

2) GRAPH REASONING MODULE

The joints of a hand are morphologically related to each other. As the distance of the joint from the palm gradually increases, the range of movement of the joint also widens. This causes a problem such as self-occlusions, which is difficult to estimate. However, we can constraint the expected space of the hand joint position by using the kinematic link information of the finger. To perform this role, we apply the graph reasoning module to incorporate the connection information between joints to the extracted features. The GRM is designed with reference to the global reasoning unit [40], and relation reasoning networks [42], [43], the detailed architecture of which is shown in Fig 3. The module can be divided into three steps.

The first step is to learn the projection function $f(\cdot)$ that transforms the convolutional feature $X \in \mathbb{R}^{H\times W\times C}$ into a new graph node feature $V = f(X) \in \mathbb{R}^{N\times C}$, where H, W, and C denote the height, width, and depth size of feature volume X, respectively, and N is the number of joint nodes in the graph space. We aim to apply relation reasoning in the graph space, which means that the joint nodes store information on the entire image. The projection weight $P = \theta(x)$ learned by the 1×1 convolution is a P_n normalized using the softmax function which uses voting to reliably assign features. In addition, we use a 1×1 convolution to generate $\phi(x)$ and reduce the input dimensions and the module parameters as follows:

$$V = P_n\,\phi(x). \tag{1}$$

In this way, different original features can adaptively vote on the representations of the joint nodes.

The second step is to capture the global relation reasoning between nodes in a graph space. Each node of V potentially stores information of the hand joints and is simplified to capture the relations between the joint nodes in the graph space. We use a fully connected layer to learn the edge weights of each node for relation reasoning. We also utilize the adjacency matrix with reference to the recently proposed GCN [36]. As shown in Fig. 3, we use an adjacency matrix that aims to spread information throughout all of the nodes. The adjacency matrix $A \in \mathbb{R}^{N\times N}$ represents the structural connection of the hand joints. We do not apply the normalized form proposed in [36] but use the parameterized adjacency matrix, which is optimized during the learning process with other parameters of the GRM. Our method learns about the connection strength as well as the inter-joint connectivity during the training process, and thus better node features can be obtained. The features updated through relation reasoning using matrix multiplication are as follows:

$$Y = \sigma\,(A\,V\,W_g), \tag{2}$$

where $\sigma(\cdot)$ is a ReLU function as a nonlinear activation function, and $W_g \in \mathbb{R}^{C' \times C'}$ is a learnable weight matrix.

The last step is to project the learned node features back to the original coordinate space. The output feature Y contains information about the relation among the hand joints. To update the features available for the following network, we apply a reverse projection to project the graph features back. We reuse the projection weight P_n created in the first step to reduce the computational cost. As shown on the right side of Fig. 3, after the features from the GCN are transferred to the original space, a 1×1 convolution is applied for the dimension expansion. The output is concatenated with the input feature X through the residual path for stabilization, and a 1×1 convolution is then added for conversion into the input dimension.

C. LOCAL FEATURE REFINEMENT AND REGRESSION

The variations of the palm are relatively stable, but the five fingers are mostly independent and highly flexible. Because each finger has a large degree of freedom and a wide activity space, it generates problems such as self-occlusions and self-similarities, which are difficult to estimate. Conventional hierarchical branch methods treat the joints included in each finger equally, which is insufficient to obtain representative features of the hand structure. Our method learns the connection information of the entire hand joints and the connection relation information of each finger joint, and then incorporates that information into the features. As shown in Fig. 2, we further improve the estimation accuracy by adding the local GRM to each finger branch to refine the local features.

Because the palm determines the main position of the hand and the variation is small compared to the flexible fingers, the palm position is first predicted through the heatmap. We then construct a hierarchical network of six separate branch networks for the estimation of the palm, thumb, index, middle, ring, and little finger, each of which is optimized. We apply the branch ensemble method to estimate the overall coordinates by concatenating features from the fully connected layers of each branch. This method can improve estimation accuracy by incorporating the correlation between fingers rather than estimating coordinates directly in each branch.

The detailed architecture of the proposed method is described in Table 1, and the number of reduced dimensions (C') in the GRM uses 128 channels which is smaller than the input feature dimension $(C = 256)$ to reduce the computational cost.

TABLE 1. Detailed architecture of the proposed method without GRM.

Network	Layout	Kernel size	# channels	Output size
Global Feature Extraction	Convolution	7 × 7	32	48
	Residual block	3 × 3	32	24
	Residual block	3 × 3	64	12
	Residual block	3 × 3	128	6
	Residual block	3 × 3	128	3
	Upsampling	1 × 1	256	6
	Upsampling	1 × 1	256	12
Local Feature Extraction	Residual block	3 × 3	256	12
	Residual block	3 × 3	256	12
	Max pooling	3 × 3	256	6
Regression	Fully-connected	-	-	1024
	Fully-connected	-	-	Joints × 3

D. LOSS FUNCTIONS

The total loss of this architecture for training is the sum of the four losses, i.e., two heat map losses and two regression losses.

The first is the heat map loss of the initial feature used as a constraint for a better global feature extraction:

$$L_{ht}^{H} = \sum_{i=1}^{A} \sum_{u}^{w} \sum_{v}^{h} \left\| H_i(u,v) - H_i^{*}(u,v) \right\|^2, \tag{3}$$

where A is the total number of joints of the hand, $H_i \in \mathbb{R}^{w \times h}$ and H_i^* denote the estimated heat map and ground truth heatmap of the i-th joint.

The second is the heatmap loss of the local feature refinement network used to extract the intermediate features of the palm and each finger.

$$L_{ht}^{j} = \sum_{i=1}^{n_j} \sum_{u}^{w} \sum_{v}^{h} \left\| H_i(u,v) - H_i^{*}(u,v) \right\|^2,$$
$$j \in \{Palm,\ Thumb,\ Index,\ Middle,\ Ring,\ Little\}, \tag{4}$$

where n_j is the number of joints in the j-th branch, and $H_i$and H_i^* represent the estimated heat map and the ground truth heatmap of the i-th joint at the j-th branch.

Third, the subtask losses are used for each branch in the regression network. The fourth loss supervises the final output of the entire hand joint. In addition, we adopt the feature ensemble method used in [15], [27].

$$L_r^{j} = \sum_{i=1}^{n_j} \left\| C_i - C_i^{*} \right\|_2^2, \tag{5}$$

$$L_r^{H} = \sum_{i=1}^{A} \left\| C_i - C_i^{*} \right\|_2^2, \tag{6}$$

where C_i and C_i^* indicate the estimated 3D joint coordinates and ground truth of the i-th joint at the j-th branch, respectively, and C_i and C_i^* denote the estimated 3D joint coordinates of the i-th joint among all joints and the corresponding ground truth. The total loss function is defined as follows:

$$L = \alpha L_{ht}^{H} + \beta(L_{ht}^{P} + L_{ht}^{T} + L_{ht}^{I} + L_{ht}^{M} + L_{ht}^{R} + L_{ht}^{L}) + \gamma(L_r^{P} + L_r^{T} + L_r^{I} + L_r^{M} + L_r^{R} + L_r^{L}) + \delta L_r^{H}, \tag{7}$$

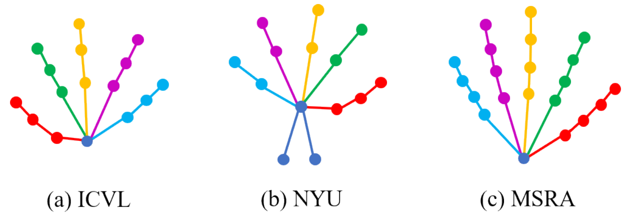

FIGURE 4. The palm joints (bule points) and the finger joints subset on three datasets. The five fingers corresponding to the local features are color-coded.

where α, β, γ, and δ are balance factors that weight the four losses, respectively. We optimally set $\alpha = 1$, $\beta = 1$, $\gamma = 0.01$, and $\delta = 0.01$ during our experiment.

E. IMPLEMENTATION DETAILS

The proposed method is implemented using the TensorFlow framework, and training and testing are conducted using an NVIDIA 2080 Ti GPU. Similar to previous studies [13], [17], [25], we crop the hand region from the original image and then resize the cropped image to a fixed pixel resolution of 96 × 96. We then normalize the depth value of the resized image to $[-1, 1]$. We use online data augmentation to improve model performance with the method proposed in [13], including random rotation, random scaling, and random translation.

We trained the model in an end-to-end manner for 100 epochs with a batch size of 32 using the RMSProp optimizer. The initial learning rate was set to 0.0005 and reduced by 0.95-fold for every epoch. To prevent an overfitting, the weight decay was set to 1e-5 and the dropout rate was set to 0.6.

IV. EXPERIMENTAL RESULTS

In this section, we briefly introduce three public datasets (NYU, ICVL, and MSRA) of hand pose estimation and describe the evaluation metrics. The following self-comparisons were conducted to verify the effectiveness of the architectural structure and graph-based reasoning module. Finally, we show the quantitative and qualitative results compared with previous state-of-the-art methods.

A. DATASETS AND EVALUATION METRICS

The ICVL dataset [21] was acquired using an Intel RealSense camera, and contains 330k frames of training data and 1.6k frames of testing data. As shown in Fig. 4(a), there is one palm joint and three joints per finger, thereby annotating a total of 16 joints. The NYU dataset [22] was captured using Microsoft Kinect at three viewpoints. This dataset contains 72k training frames and 8.2k testing frames with annotations for 36 joints, as shown in Fig. 4(b). We used 14 joints among 36 joints in the frontal view to evaluate in the same way as the previous methods. The MSRA dataset [23] was generated using Microsoft Research Asia and was obtained using Intel's gesture camera. This dataset contains 76.5k images divided into nine subjects. As shown in Fig. 4(c), a total of 21 joints consisting of one palm joint and four joints of each finger was annotated. For comparison with other methods, we evaluated using the same leave-one-subject-out cross-validation strategy.

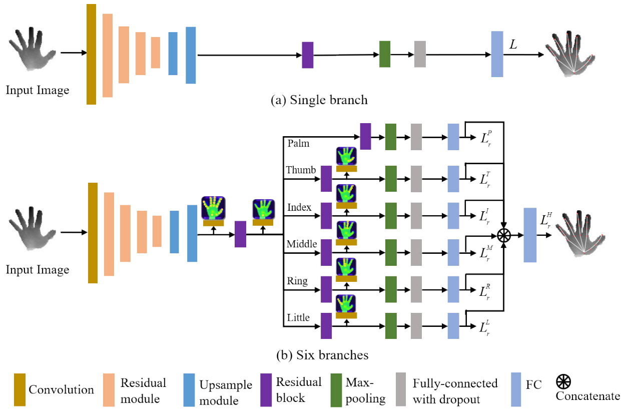

FIGURE 5. Baseline architectures for ablation study.

TABLE 2. Comparison of ablation study on three hand pose datasets.

Model	Mean error (mm)		
	ICVL	NYU	MSRA (S0)
Baseline (single branch)	6.79	9.42	7.53
Single branch + Global GRM	6.51	9.12	7.41
Baseline (six branches)	6.58	8.95	7.42
Six branches + Global GRM	6.46	8.78	7.38
Six branches + Local GRM	6.31	8.81	7.30
Six branches + Both GRMs	6.25	8.57	7.22

We used two common metrics to compare with other state-of-the-art methods. One is the average 3D distance error, which is the Euclidean distance between the predicted joint coordinates and ground truth joint coordinates. The other metric is the proportion of test frames whose joint errors are within a threshold [47].

B. ABLATION STUDY

We conducted various ablation experiments on three public datasets as shown in Table 2. To evaluate the performance of the proposed architecture, we compared the experimental results with two baseline architectures, as shown in Fig.5. One of the baseline architectures has a single branch structure, and the other has a hierarchical six-branch structure without a global or local GRM. During this experiment, we selected and evaluated the first subject S0 out of nine subjects of the MSRA dataset as a validation subject.

1) HIERACHICAL MULTI-BRANCH ARCHITECTURE

A hand pose estimation is an extremely complex task, and regressing all joints directly together is ineffective. Dividing a complex task into several sub-tasks makes each more generalized. Therefore, we improved the estimation accuracy by regressing six branches with one palm and five fingers in a structure similar to the methods described in [15] and [26]. On the NYU dataset, the distance mean error was significantly reduced from 9.42 to 8.95 mm, and the comparison

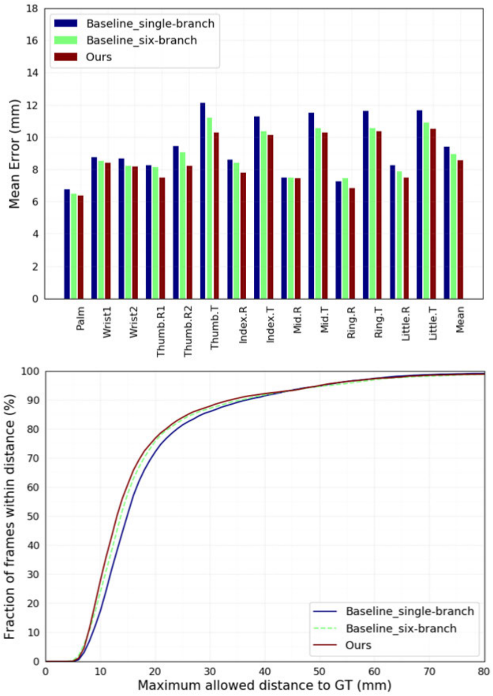

FIGURE 6. **Self-comparison results on NYU dataset. Upper: 3D distance errors (mm) per hand joints. Lower: Percentage of success frames over different error thresholds.**

results are shown in Fig. 6 and Table 2. Experiments on the ICVL and MSRA datasets also show improved accuracy when applying the six branch structures.

2) GRAPH REASONING MODULE

We also evaluated the effect of the GRM on finger joint estimation. As shown in Table 2, adding a global GRM to the single branch baseline architecture reduced the mean error of the hand joints from 9.42 mm to 9.12 mm for the NYU dataset. In addition, when adding a global GRM and local GRMs to the six branches baseline architecture, the mean errors decreased from 8.95 mm to 8.78 mm and 8.81 mm, respectively. This is because the global GRM improved overall hand characteristics and the local GRM improved each finger characteristic. When both GRMs is applied, the performance was significantly improved to 8.57mm. The other two datasets show the same tendency in that the mean error distance decreased when GRMs are added.

C. COMPARISON WITH STATE-OF-THE-ART METHODS

The performance of the proposed method was compared with previous state-of-the-art methods using a 2D or 3D depth image input. The 3D input models are a 3DCNN [16], SHPR-Net [29], HandPointNet [30], Point-to-Point [31], and V2V-PoseNet [14], and the 2D input models are a DeepModel [19], Deep-Prior [12], Deep-Prior++ [13], Feedback [32], REN-4 × 6 × 6 [25], REN-9 × 6 × 6 [33], Pose-REN [17], CrossInfoNet [28], HCRNN [34], and A2J [35].

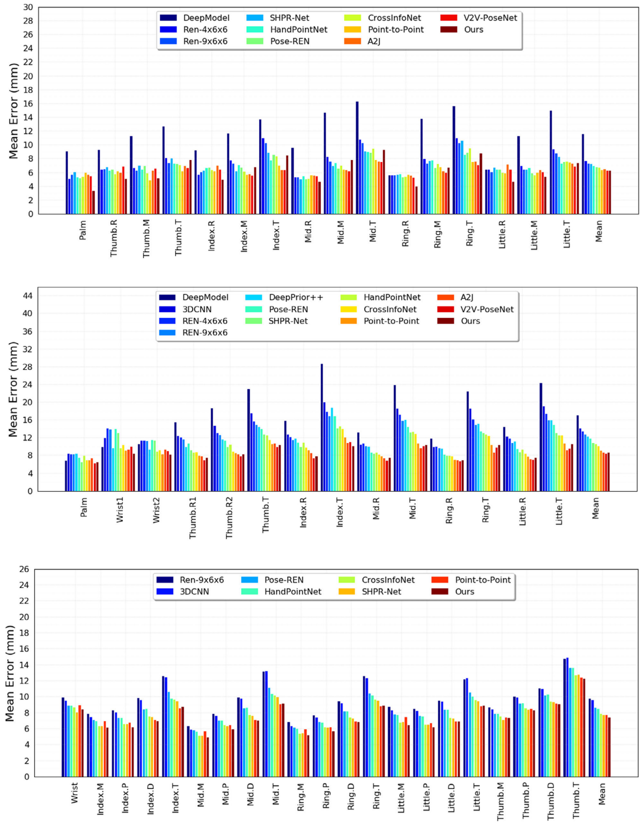

FIGURE 7. **Comparison with state-of-the-art methods for 3D distance errors per hand joints. Top: ICVL [21] dataset. Middle: NYU [22] dataset, Bottom: MSRA [23] dataset.**

The results of the average 3D distance error per hand joint and the successful frame rate on the three datasets are shown in Fig. 7, Fig. 8, and Table 3. Our proposed method shows a 0.03- and 0.19-mm better performance than the V2V [14] method on the ICVL and MSRA datasets, respectively, and achieved the best performance on all methods. On the NYU dataset, our method is the second most accurate. However, as shown in the middle of Fig. 8, the percentage of successful frames of our method shows a good performance above a threshold of 10 mm. Although the accuracy of our method and the V2V approach [14] is similar, the V2V of the 3D input uses 3D CNNs with a high computational cost. Therefore, the inference speed of V2V is extremely slow at 3.5 frames per second (fps), which makes it difficult to utilize in real-time systems. We additionally experimented with the average 3D distance error distributed over various yaw and pitch angles on the MSRA dataset as shown in Fig. 9. Our method achieved superior results in almost all of the yaw and pitch angles, which indicates the robustness of our method to viewpoint changes. The qualitative results of the proposed method on the three datasets are shown in Fig. 10, where the yellow line represents the ground truth of the joint coordinates and the red line shows our prediction results.

We evaluated the estimated speed of our proposed method on an Intel Core i7 CPU, 32GB RAM, and an

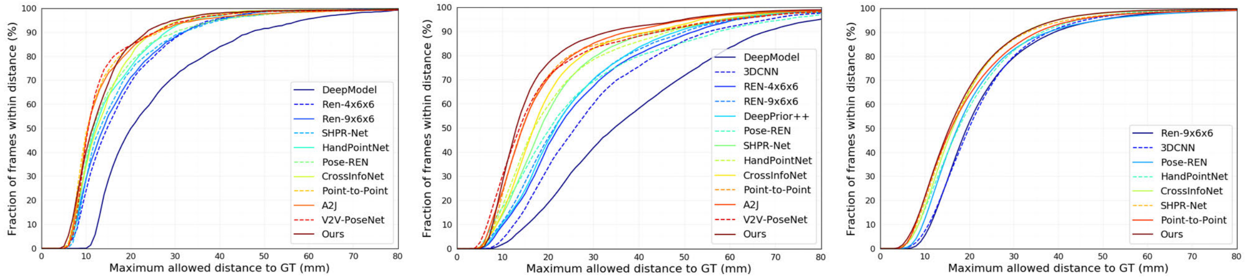

FIGURE 8. Comparison with state-of-the-art methods for percentage of successful frames over different error thresholds. Left: ICVL [21] dataset, Middle: NYU [22] dataset, Right: MSRA [23] dataset.

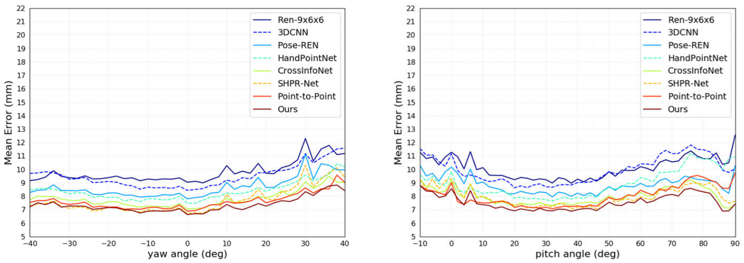

FIGURE 9. Comparison of mean 3D distance error distributed over different yaw (left) and pitch (right) viewpoint angles on the MSRA [23] dataset.

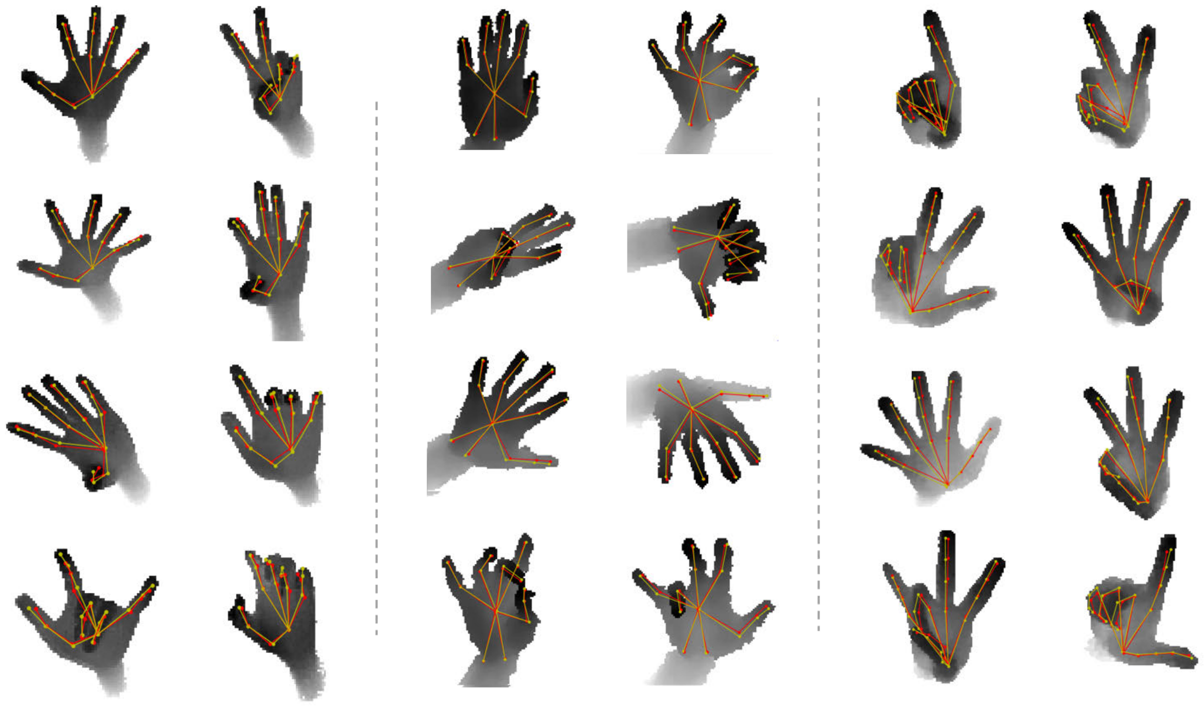

FIGURE 10. Qualitative results using our method on three public hand pose datasets. We compare our method with the ground truth joint locations. The predicted results are in red and the ground truth are in yellow. Left: ICVL [21] dataset, Middle: NYU [22] dataset, Right: MSRA [23] dataset.

NVIDIA 2080ti single GPU environment. Our method runs at an inference speed of 103 fps. Because the maximum frame rate of recent consumer depth cameras is 60 fps (or 90 fps), our method can be used as a real-time application. If real-time operation is required in a resource-limited environment, such as an embedded system, the single branch and global GRM method can be considered. This method has a slight decrease in accuracy (NYU 9.12 mm) but can be operated quickly at 165 fps with a GPU.

TABLE 3. Comparison of the proposed method with state-of-the-art methods on three hand pose estimation datasets. Mean error indicates the average 3d distance error over all joints. fps indicates frames per second.

Methods	Mean error (mm)			Input	fps
	ICVL	NYU	MSRA		
3DCNN [16]	-	14.1	9.58	3D	-
SHPR-Net [29]	7.22	10.78	7.76	3D	-
HandPointNet [30]	6.94	10.5	8.5	3D	48
Point-to-Point [31]	6.33	9.04	7.71	3D	41.8
V2V-PoseNet [14]	**6.28 (2)**	**8.42 (1)**	**7.59 (2)**	3D	3.5
DeepModel [19]	11.56	17.04	-	2D	-
DeepPrior [12]	10.4	19.73	-	2D	-
DeepPrior++ [13]	8.1	12.24	9.5	2D	30
Feedback [32]	-	15.97	-	2D	-
REN-4x6x6 [25]	7.63	13.39	-	2D	-
REN-9x6x6 [33]	7.31	12.69	9.79	2D	-
Pose-REN [17]	6.79	11.81	8.65	2D	-
CrossInfoNet [28]	6.73	10.08	7.86	2D	124.5
HCRNN [34]	6.54	9.37	7.7	2D	285
A2J [35]	6.46	8.61	-	2D	105.1
Ours	**6.25 (1)**	**8.57 (2)**	**7.4 (1)**	2D	103

V. CONCLUSION

In this paper, we proposed a novel approach for 3D hand pose estimation from a single depth image using relation reasoning between hand joints by projecting the coordinate space information to nodes in a graph space. This graph reasoning module (GRM) is a lightweight and easy-to- optimize block that performs projection and reverse projection, and provides features including learned spatial connection information between hand joints using graph convolutional networks. We also designed an architecture that applies local GRMs to finger branches to add relation information for each finger joint. In the experimental results, we conducted various ablation experiments to demonstrate the effectiveness of the hierarchical architecture of the proposed method and application of the GRM. Our proposed method achieved the highest and most promising estimation accuracy compared with previous state-of-the-art methods. Our method was outstanding in terms of accuracy and efficiency because it not only had the highest accuracy on three public datasets, but also enabled real-time operation at a relatively high speed of 103 fps. In the future, we will study more efficient network that can operate smoothly even in resource-limited environments.

REFERENCES

[1] T. Piumsomboon, A. Clark, M. Billinghurst, and A. Cockburn, "User-defined gestures for augmented reality," in *Proc. CHI Extended Abstr. Hum. Factors Comput. Syst. (CHI EA)*, 2013, pp. 955–960.

[2] Y. Jang, S.-T. Noh, H. J. Chang, T.-K. Kim, and W. Woo, "3D finger CAPE: Clicking action and position estimation under self-occlusions in egocentric viewpoint," *IEEE Trans. Vis. Comput. Graphics*, vol. 21, no. 4, pp. 501–510, Apr. 2015.

[3] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel(R) RealSense(TM) stereoscopic depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1–10.

[4] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE MultimediaMag.*, vol. 19, no. 2, pp. 4–10, Feb. 2012.

[5] S. Yuan *et al.*, "Depth-based 3D hand pose estimation: From current achievements to future goals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2636–2645.

[6] A. Farooq and C. S. Won, "A survey of human action recognition approaches that use an RGB-D sensor," *IEIE Trans. Smart Process. Comput.*, vol. 4, no. 4, pp. 281–290, Aug. 2015.

[7] W. Chen, C. Yu, C. Tu, Z. Lyu, J. Tang, S. Ou, Y. Fu, and Z. Xue, "A survey on hand pose estimation with wearable sensors and computer-vision-based methods," *Sensors*, vol. 20, no. 4, p. 1074, Feb. 2020.

[8] S. Khamis, J. Taylor, J. Shotton, C. Keskin, S. Izadi, and A. Fitzgibbon, "Learning an efficient model of hand shape variation from depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2540–2548.

[9] D. J. Tan, T. Cashman, J. Taylor, A. Fitzgibbon, D. Tarlow, S. Khamis, S. Izadi, and J. Shotton, "Fits like a glove: Rapid and reliable hand shape personalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5610–5619.

[10] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3213–3221.

[11] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. W. Fitzgibbon, and S. Izadi, "Accurate robust and flexible real-time hand tracking," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, Apr. 2015, pp. 3633–3642.

[12] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," in *Proc. Comput. Vis. Winter Workshop*, 2015, pp. 1–10.

[13] M. Oberweger and V. Lepetit, "DeepPrior++: Improving fast and accurate 3D hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 585–594.

[14] J. Y. Chang, G. Moon, and K. M. Lee, "V2 V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5079–5088.

[15] M. Madadi, S. Escalera, X. Baró, and J. Gonzalez, "End-to-end global to local CNN learning for hand pose recovery in depth data," 2017, *arXiv:1705.09606*. [Online]. Available: http://arxiv.org/abs/1705.09606

[16] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Real-time 3D hand pose estimation with 3D convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 956–970, Apr. 2019.

[17] X. Chen, G. Wang, H. Guo, and C. Zhang, "Pose guided structured region ensemble network for cascaded hand pose estimation," *Neurocomputing*, vol. 395, pp. 138–149, Jun. 2020.

[18] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Crossing nets: Combining GANs and VAEs with a shared latent space for hand pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 680–689.

[19] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei, "Model-based deep hand pose estimation," in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 2421–2427.

[20] Q. Ye, S. Yuan, and T.-K. Kim, "Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Cham, Switzerland: Springer, 2016, pp. 346–361.

[21] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3D articulated hand posture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3786–3793.

[22] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, p. 169, Sep. 2014.

[23] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 824–832.

[24] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3D hand pose estimation in single depth images: From single-view CNN to multi-view CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3593–3601.

[25] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang, "Region ensemble network: Improving convolutional network for hand pose estimation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4512–4516.

[26] A. Sinha, C. Choi, and K. Ramani, "DeepHand: Robust hand pose estimation by completing a matrix imputed with deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4150–4158.

[27] Y. Zhou, J. Lu, K. Du, X. Lin, Y. Sun, and X. Ma, "HBE: Hand branch ensemble network for real-time 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 2018, pp. 501–516.

[28] K. Du, X. Lin, Y. Sun, and X. Ma, "CrossInfoNet: Multi-task information sharing based hand pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9896–9905.

[29] X. Chen, G. Wang, C. Zhang, T.-K. Kim, and X. Ji, "SHPR-net: Deep semantic hand pose regression from point clouds," *IEEE Access*, vol. 6, pp. 43425–43439, 2018.

[30] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand PointNet: 3D hand pose estimation using point sets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8417–8426.

[31] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 475–491.

[32] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3316–3324.

[33] G. Wang, X. Chen, H. Guo, and C. Zhang, "Region ensemble network: Towards good practices for deep 3D hand pose estimation," *J. Vis. Commun. Image Represent.*, vol. 55, pp. 404–414, Aug. 2018.

[34] C.-H. Yoo, S. Ji, Y.-G. Shin, S.-W. Kim, and S.-J. Ko, "Fast and accurate 3D hand pose estimation via recurrent neural network for capturing hand articulations," *IEEE Access*, vol. 8, pp. 114010–114019, 2020.

[35] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. T. Zhou, and J. Yuan, "A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 793–802.

[36] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. ICLR*, 2014, pp. 1–14.

[37] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. NeurIPS*, 2016, pp. 3844–3852.

[38] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017, pp. 1–14.

[39] X. Wang and A. Gupta, "Videos as space-time region graph," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 399–417.

[40] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 433–442.

[41] R. Wang, C. Huang, and X. Wang, "Global relation reasoning graph convolutional networks for human pose estimation," *IEEE Access*, vol. 8, pp. 38472–38480, 2020.

[42] Y. Li and A. Gupta, "Beyond grids: Learning graph representations for visual recognition," in *Proc. NeurIPS*, 2018, pp. 9225–9235.

[43] X. Liang, Z. Hu, H. Zhang, L. Lin, and E. P. Xing, "Symbolic graph reasoning meets convolutions," in *Proc. NeurIPS*, 2018, pp. 1853–1863.

[44] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3595–3603.

[45] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12026–12035.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[47] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon, "The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 103–110.

JAE-HUN SONG received the B.S. and M.S. degrees from Hanyang University, South Korea, in 2001 and 2003, respectively, both in electronic engineering. He is currently pursuing the Ph.D. degree in electronic engineering with Sogang University. He is also working in circuit design and IT product development at LG Display Company Ltd., South Korea. His current research interests include image processing, computer vision, and deep learning.

SUK-JU KANG (Member, IEEE) received the B.S. degree in electronic engineering from Sogang University, South Korea, in 2006, and the Ph.D. degree in electrical and computer engineering from the Pohang University of Science and Technology, in 2011.

From 2011 to 2012, he was a Senior Researcher with LG Display, where he was a Project Leader for resolution enhancement and multi-view 3D system projects. From 2012 to 2015, he was an Assistant Professor of electrical engineering with Dong-A University, Busan. He is currently an Associate Professor of electronic engineering with Sogang University. His current research interests include image analysis and enhancement, video processing, multimedia signal processing, circuit design for display systems, and deep learning systems. He was a recipient of the IEIE/IEEE Joint Award for Young IT Engineer of the Year, in 2019.